**JavaScript - Day -1: Introduction to Browser & web**

**1. Write a blog on Difference between HTTP 1.1 vs HTTP2**

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP 1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

**HTTP 1.1:**

Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web, HTTP is a top-level application protocol that exchanges information between a client computer and a local or remote web server. In this process, a client sends a text-based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client. Since early 1996, major web browsers and web server developers also started to implement new features specified by pre-standard HTTP/1.1 drafts specifications. HTTP 1.1 used to process text commands to complete request-response cycles. These are the drawbacks that lead to the creation of HTTP/2: The first problem is HTTP/1.1 transfer all the requests & responses in the plain text message form. The second one is head of line blocking in which TCP connection is blocked all other requests until the response does not receive. all the information related to the header file is repeated in every request.

**HTTP2:**

HTTP/2 began as the SPDY protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization. This protocol served as a template for HTTP/2 when the Hypertext Transfer Protocol working group httpbis of the IETF (Internet Engineering Task Force) put the standard together, culminating in the publication of HTTP/2 in May 2015. From the beginning, many browsers supported this standardization effort, including Chrome, Opera, Internet Explorer, and Safari. Due in part to this browser support, there has been a significant adoption rate of the protocol since 2015, with especially high rates among new sites. HTTP/2 uses HPACK which is used to split data from header. it compresses the header. The server sends all the other files like CSS & JS without the request of the client using the PUSH frame.

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP2** |
| It works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |

**2. Write a blog about objects and its internal representation in Javascript**

JavaScript, being a powerful and versatile programming language, offers various data structures to store and manipulate data. One of the most fundamental and widely used data structures in JavaScript is the object. Objects in JavaScript are unique entities that allow developers to represent complex data and organize it in a structured manner. In this blog, we will delve into the internal representation of objects in JavaScript, exploring their key features and providing insights into their practical usage.

**Understanding Objects in JavaScript:**

In JavaScript, an object is an unordered collection of key-value pairs, where each key is a unique string (or symbol) and each value can be of any type: primitive data types, functions, or even other objects. Objects provide a flexible way to structure and organize related data, making them a central concept in JavaScript programming.

**Internal Representation of Objects:**

Internally, JavaScript objects are implemented as dictionaries or hash tables, also known as associative arrays. This implementation choice ensures efficient retrieval and manipulation of key-value pairs. The keys within an object are stored as strings, allowing quick lookups and seamless property access.

Every object has some property associated with some value. These values can be accessed using these properties associated with them.

var myCar = new Object();

myCar.make = 'Suzuki';

myCar.model = 'Altros';

myCar.year = 2020;

myCar.wheels = 4;

After creating myCar object, the value inside the object can be accessed using keys.

i.e..

myCar.year

Output: 2020

Creating Objects in JavaScript:

1.By object literal

2.By creating instance of Object directly (using new keyword)

By object literal:

The syntax of creating object using object literal is given below:

object={property1:value1,property2:value2……propertyN:valueN}

Property and value is separated by colon(:).

**Example:**

var person={

fname:"xxx",

name:"yyy",

age:25

};

By creating instance of Object directly (using new keyword):

The syntax of creating object directly is given below:

var objectname=new Object();

Here, new keyword is used to create object.

**Example:**

var emp=new object();

emp.id=101;

emp.name="xxx";

emp.salary=50000;

Accessing JavaScript Objects:

The syntax for accessing the property of an object is:

objectName.property

or

objectName[“property”]

Accessing ‘fname’ from example 1 using dot operator,

person.fname

Accessing ‘name’ form example 2 using [],

emp[“name”]